



Gregory, J. R., Berthoud, L., Tryfonas, T., & Faure, L. (2020). *There's no 'I' in SEAM - An Interim Report on the 'Spacecraft Early Analysis Model'*. Paper presented at IEEE Aerospace Conference, Big Sky, Montana, United States.

Peer reviewed version

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

There's no 'I' in SEAM – An Interim Report on the 'Spacecraft Early Analysis Model'

Joe Gregory
Dept. of Aerospace
Engineering
University of Bristol,
University Walk,
Bristol, BS81TR
UK
joe.gregory@bristol.ac.uk

Lucy Berthoud
Dept. of Aerospace
Engineering
University of Bristol,
University Walk,
Bristol, BS81TR
UK
lucy.berthoud@bristol.ac.uk

Theo Tryfonas
Dept. of Civil
Engineering
University of Bristol,
University Walk,
Bristol, BS81TR
UK
theo.tryfonas@bristol.ac.uk

Ludovic Faure
Airbus Defence and Space
Gunnels Wood Road
Stevenage, SG12AS
UK
ludovic.faure@airbus.com

Abstract—Model-Based Systems Engineering (MBSE) represents a move away from the traditional approach of Document-Based Systems Engineering (DBSE), and is used to promote consistency, communication, clarity and maintainability within systems engineering projects. In previous work, industry focus groups have indicated that one way this can be achieved is by performing early functional validation of elements of the spacecraft avionics.

This paper presents an extended approach, introduced in a case study previously published by the authors, to enable early functional analysis of a spacecraft. The approach uses the 'Spacecraft Early Analysis Model' (SEAM), a SysML-based model framework for the definition, development and analysis of a space-based mission and corresponding space system. This formal model-based representation of the system enables the high-level simulation of the design during Phase B of the spacecraft system lifecycle.

The SEAM pulls together different, traditionally disparate, analysis tools and enables them to work together, producing an integrated system model spanning multiple tools. It facilitates the simulation of the mission using dedicated orbit modelling software, analysis of the completeness and accuracy of the system behaviour, and provides an indication of the appropriate logical architecture.

The SEAM has been developed iteratively by applying it to Earth-observation case studies from the Biomass mission, refining the capabilities of the template accordingly, and subsequently generalising the model. The resulting interim version of the Spacecraft Early Analysis Model contains a series of MBSE patterns that will ultimately provide users with a comprehensive and consistent SysML-based structure that enables early functional definition and analysis of spacecraft.

Next steps in the development of the SEAM include its application to a wider variety of use cases to develop and demonstrate its versatility, and the development of metrics to measure its perceived value among practitioners.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. MBSE AND SPACECRAFT	2
3. METHODOLOGY	3
4. SPACECRAFT EARLY ANALYSIS MODEL	4
5. DISCUSSION	11
6. NEXT STEPS	12
7. CONCLUSIONS	12
ACKNOWLEDGEMENTS	12
REFERENCES	12
BIOGRAPHY	14

1. INTRODUCTION

Interest in Model-Based Systems Engineering (MBSE) over the traditional approach to systems engineering, Document-Based Systems Engineering (DBSE), is growing [1], [2]. With DBSE, project and design information is stored in documents and must be manually maintained and transferred between domains [3], [4]. The traditional DBSE approach can be labour-intensive and consists mostly of manual upkeep, review and inspection [5].

MBSE is the formalised application of modelling to support system requirements, design, analysis, optimisation, verification and validation [6]. By using interconnected models to store, represent and relate this information and data, projects can expect improvements in consistency, communication, clarity, visibility, maintainability, etc. – thus addressing issues associated with cost, complexity and safety [7].

Spacecraft represent an ideal candidate for the application of MBSE as they are complex systems with potential applications that are often limited by the high development costs they can incur [8], [9].

In previous work, the authors have presented an extended approach, first described in a case study published in [10], to enable early functional definition and analysis of a spacecraft. The case study described forms part of a larger work effort to use MBSE techniques to develop a model-based template that is capable of describing and simulating a space-based mission and corresponding spacecraft system at a high-level during the early design phases. This work focuses on ‘Phase B’, an early phase of the spacecraft system lifecycle – the aim of Phase B is to establish a functionally complete preliminary design solution [11].

The goals of this work are as follows:

- Develop a Spacecraft Early Analysis Model (SEAM) to be used as a template for early functional definition and analysis of spacecraft.
- Demonstrate the applicability and flexibility of the SEAM by applying it to real space-based missions under development at Airbus.
- Investigate the benefits (qualitatively or quantitatively) of using this approach rather than traditional DBSE techniques on real projects.

In this paper, the current state of development of this model template structure, the SEAM, is presented. In its current state, the SEAM comprises an innovative model structure that facilitates system-level simulation and analysis against the mission needs. It achieves this by maintaining separation between the mission and the system and using the mission profile to drive a simulation of the system response. The system functionality is described by modular MBSE patterns. Included in this overview are these MBSE patterns, to be followed when applying the SEAM to a specific spacecraft mission. The methodology that has been followed, resulting in the SEAM’s current status, and the proposed future direction of the SEAM’s development are also described.

2. MBSE AND SPACECRAFT

MBSE provides the opportunity to link various domain-specific tools together to produce a model-based framework for a systems engineering project. It is often discussed in terms of the three MBSE pillars: language, tool and methodology [12]. The tool is the software used to produce the model, which consists of model elements, tables, diagrams, etc. representing the appropriate modelling language. Of the multiple languages available [13], the Object Management Group’s (OMG) Systems Modeling Language (SysML) has become the de facto modelling language for systems engineering [14], and is well suited to the description of the MBSE activities [15]. The methodology is the process used to build the model.

There have been multiple space-mission-based MBSE workstreams undertaken within Airbus. The Jupiter ICy moons Explorer (JUICE) mission, for example, used a

model-based approach to system optimisation in terms of instrument parameters, mass storage configuration and transmission band allocation [16].

The most comprehensive and detailed effort so far, however, is the application of a model-based process to support the iterative generation and maturation of the system requirements, architectures and system budgets of the e.Deorbit mission [17]. The e.Deorbit mission is an ESA-led project with the aim of ‘removing a single large ESA-owned space debris from the low-Earth orbit protected zone’ [18]. It underwent the application of the ‘Federated and Executable Models’ approach, developed by Estable [17], [19], and benefitted from automated trade studies spanning multiple tools, a clear distinction between the mission (the needs) and the system (the solution), and explicit traceability between design features and requirements. The approach’s corresponding SysML-based template has linked SysML, RHEA’s Concurrent Design Platform [20], MathWorks’ MATLAB [21] and Phoenix Integration’s ModelCenter [22] and RangeDB (a database tool developed by Airbus) in one process for implementing the design-analysis-verification workflow [23].

The Federated and Executable Models template has been used as a basis for the development of the SEAM and as such the underlying structure remains largely similar. In particular, the key concept of distinction between the mission and the system remains in place. These structures and patterns are presented in greater detail in Section 5. The Federated and Executable Models template, however, is limited in its ability to perform a comprehensive mission-level simulation that would analyse the system functionality in terms of its ability to address the mission needs. This model-based approach also relies on ModelCenter to link together multiple engineering tools and enable cross-platform analysis.

Another example of a SysML-based model framework for spacecraft is the CubeSat Reference Model (CRM), developed by the International Council on Systems Engineering (INCOSE) Space Systems Working Group (SSWG), led by David Kaslow, and is intended to be used by university project teams. The CRM aims to provide a template by which MBSE techniques can be applied to a CubeSat mission. Through collaborations with universities this has resulted in a comprehensive and intuitive CRM and a wealth of publications [9], [24]–[27].

Of particular interest is the application of the CRM to the Radio Aurora Explorer (RAX) CubeSat mission [26], which develops the analysis capabilities from static parametric representations of the system to analyses of the system evolution over time. The RAX model contained multiple state machines describing the system and activity diagrams that can trigger transitions within these state machine diagrams on execution, thus providing the evolution of these states over time. Similarly to the Federated and Executable Models approach previously described, ModelCenter has been used to integrate multiple analysis tools.

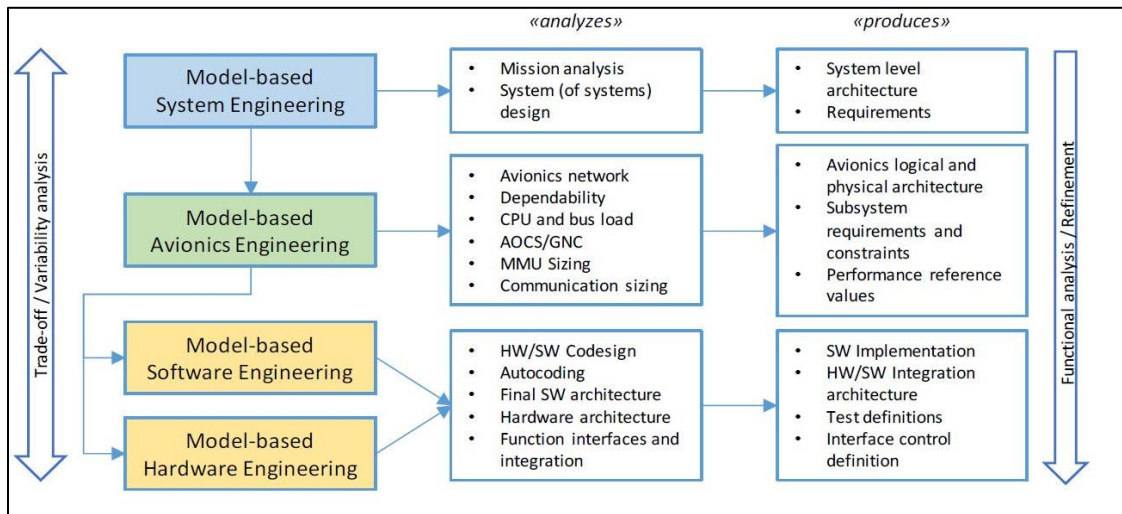


Figure 1: Context of Model-Based Avionics Engineering (MBAE) [31]

While there have been efforts to develop the MBSE approach or the simulation and analysis of spacecraft, therefore, the general focus remains on the description of system designs and often stops short of using the information present in the model to automatically analyse and validate the system itself [28], [29]. MBSE makes this possible in early phases [30].

The unique features of the SEAM are that it develops the simulation capabilities of a SysML-based modelling approach. As early as Phase B in the spacecraft system lifecycle, there may exist information regarding the mission phases, concept of operations, system / subsystem modes and logical architecture that is of sufficient maturity to perform a high-level simulation of the system functionality in terms of the mission needs. To achieve this, a model template is required that structures this information in such a way as to enable its execution. The SEAM aims to provide this capability. It is critical, however, that the approach adopted to enable the analysis of this information does not jeopardise the clarity of the information – the integrity of the information in terms of its communicability, consistency and clarity must be maintained.

3. METHODOLOGY

In this section, the methodology that has been used to determine the specific focus of the SEAM and enable its subsequent development is described. The project is driven by the Spacecraft Functional Avionics domain of Airbus and as such the initial broad objective was to apply MBSE techniques to the design and development of spacecraft functional avionics. Functional Avionics is concerned with the functionality of the spacecraft only – not the physical implementation of this functionality.

In this sense, this project focusses on the application of Model-Based Avionics Engineering (MBAE), rather than MBSE, as it is restricted to the functional avionics of the spacecraft. This level of abstraction is presented in Figure 1 as defined by the European Space Agency (ESA) [31]. MBAE looks to apply the same MBSE techniques to this

restricted view of the system – doing so with avionics-focussed case studies. Functional Avionics comprises the following domains:

- Operations
- Failure, Detection, Isolation and Recovery (FDIR)
- Software
- Attitude, Orbit Control System / Guidance Navigation and Control (AOCS/GNC)
- Database
- Functional Verification

The AOCS/GNC and Database domains have already received considerable attention within Airbus in terms of MBSE development, and the Functional Verification domain concerns the development of test beds and test procedures for Phase D, a specialised domain not of relevance here. The domains of Operations, FDIR and Software are therefore the most interesting from an Airbus perspective.

Interviews

In order to determine the particular domain(s) within Functional Avionics that was most in need of MBSE support, nine interviews were conducted – involving a total of 25 Airbus engineers working in Functional Avionics. The objective of these interviews was to identify where the current issues with the existing systems engineering processes lie, and where a model-based approach may be able to help, from the perspective of these engineers. The semi-structured interviews were conducted with engineers working in Operations, Software and FDIR. The acquired data was thematically analysed to extract common themes from the responses. This work is presented in detail in a separate paper [32].

The results of this work yielded four recommended application areas to consider when applying MBSE to Functional Avionics: organisation modelling; early functional validation; communication and consistency;

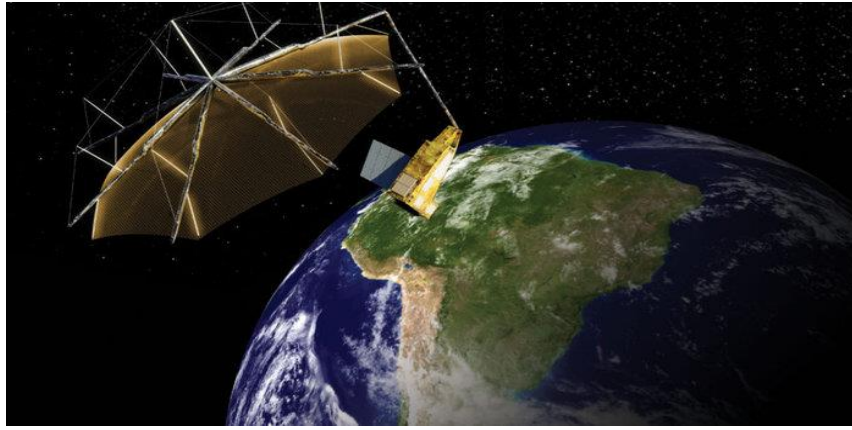


Figure 2: The Biomass Spacecraft [38]

template model framework development. This feedback supported the need for the development of a model-based template and highlighted possible use cases, in particular the early validation of the concept of operations.

Biomass Use Cases

Narrowing the focus of the work from ‘MBSE techniques applied to Functional Avionics’ to ‘early validation of the concept of operations’ enabled the consideration and selection of specific use cases. The SEAM was to be principally ‘developed by use case’ – whereby the template is developed and features are added to accommodate the needs of a particular use case, thus creating a specific mission model, and then re-generalised on completion of the use case to produce the next version of the template, complete with these new features. Each subsequent use case follows the same process and therefore each time the SEAM is applied to a use case the two results are 1) simulation outputs useful to the use case itself and 2) an updated version of the SEAM.

Two use cases have so far been identified. Both of these use cases are derived from the ESA Biomass mission. The Biomass mission is an Earth-observation mission due to be launched around 2022. The primary mission objectives are to determine the distribution of above-ground biomass in the world forests and to measure annual changes in this stock over the period of the mission [33], [34]. To achieve these objectives, a P-band (435 MHz) Synthetic Aperture Radar (SAR), making use of a deployable reflector, has been selected as the payload.

The Biomass space segment consists of a single low-Earth orbit spacecraft (Biomass) carrying the SAR instrument and reflector (Figure 2). The mission will provide global coverage twice per year over the five-year mission. To achieve this, the spacecraft will be put into a sun-synchronous orbit during the nominal operations phase. A near-repeating ground track with a period of three days will be used – a combination of controlled westward drift (of a small percentage of the instrument swath per orbit), rolling manoeuvres to position the SAR instrument and orbit drift phases are used to ensure that the global coverage requirement can be achieved [34].

The first Biomass use case concerns the definition and simulation of the mass memory onboard the Biomass spacecraft, and the functionality by which the P-band SAR data and housekeeping data are recorded, stored onboard and downlinked to Earth. Requirements concerning the physical architecture were derived by simulating the proposed system functionality and assessing against the mission needs. This work is detailed in [10]. The resulting mission-specific model has also been evaluated in terms of its flexibility and robustness to some examples of changes common to systems engineering projects [35].

The second Biomass use case concerns the early analysis of critical sequences, particularly the initialisation and deployment sequences, and adds more detail to the concept of the ‘ground station’ to which telemetry is sent and from which telecommands are received.

4. SPACECRAFT EARLY ANALYSIS MODEL

The SEAM is presented in its current state and its key structural features are noted. Its evolution from the original Federated and Executable Models template structure is also noted in the context of its application to the two use cases defined in the previous section. The SEAM is centred on a core SysML-based model created with Cameo Systems Modeler 19.0 [36]. This core model comprises the following sub-models:

- Mission Profile
- Life Cycle Stages and Mission Phases
- Functional Architecture
- Logical Architecture
- External Entities

The core model is connected to other definition and analysis tools (MATLAB, Microsoft Excel and AGI Systems Tool Kit (STK) [37]), and pulls these traditionally independent tools together to produce an integrated system model. The overall structure of the SEAM is presented in Figure 3. Overviews of these aspects of the SEAM are provided and the simulation sequence is subsequently detailed.

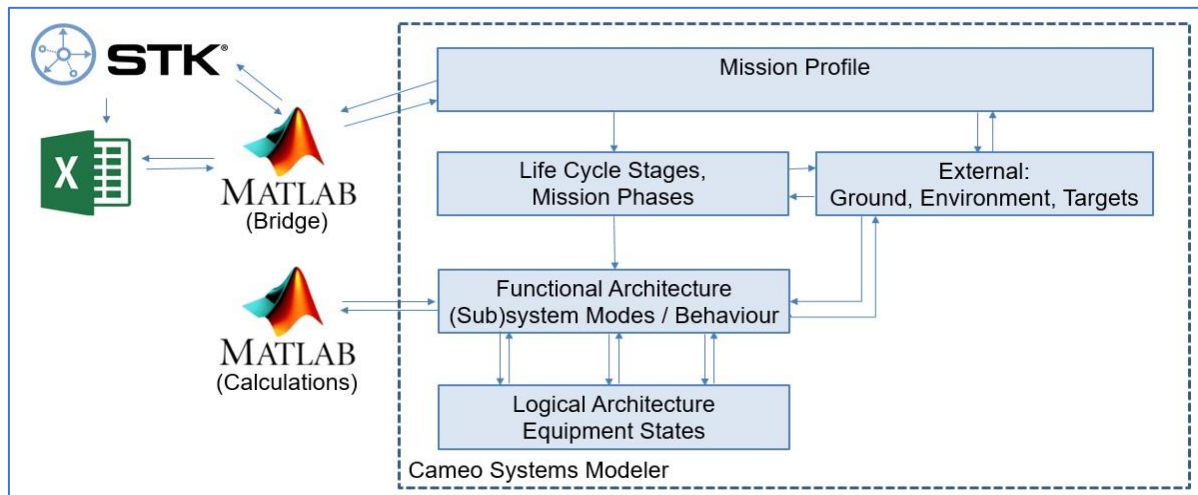


Figure 3: Spacecraft Early Analysis Model Structure (the arrows represent the flow of information on execution)

Life Cycle Stages and Mission Phases

The separation between ‘mission’ and ‘system’ is a key concept in the development of the SEAM that has descended directly from the Federated and Executable Models approach. In this sub-model, the mission itself is defined. Life cycle stages refer to the spacecraft life cycle from ‘Implementation’ onwards and so include such stages as ‘In Testing’, ‘In Operations’ and ‘In Closeout’. Each life cycle stage may then be composed of multiple phases. The Life Cycle Stages diagram, presented as a SysML state machine, is displayed in Figure 4. Figure 5 presents an example of phases – the Mission Operational Phases diagram owned by the ‘In Operations’ life cycle stage. It is worth reiterating that the life cycle stages and phases **do not** describe the system itself – they are the stages and phases that the system will experience and therefore define the mission needs that the system must be designed to meet.

Mission Profile

The Mission Profile acts as the SysML-based interface between the dedicated mission analysis tool, STK, and the rest of the core model. It contains the relevant orbit definitions (in terms of their orbital elements) and is able to drive the mission analysis via MATLAB.

Functional Architecture

The Functional Architecture contains the system functionality that has been designed to meet the needs of the mission. It is presented as the decomposition of a small number of critical, high-level functions into a large number of low-level functions. A mode can be defined as a set of functions, grouped in such a way that each mode has the functionality available to meet the needs of a particular mission phase. The System Modes diagram is presented in Figure 6 and contains the general versions of modes common to a space mission. As an example, the system mode ‘Operations Mode 1’ will contain the functionality necessary to meet the needs of the mission phase ‘Ops Phase 1’, seen in

Figure 5. This functionality can then be allocated to elements of the logical architecture.

Logical Architecture

The Logical Architecture sub-model contains a preliminary, generalised logical architecture of the system – which is assumed to be a single spacecraft (Figure 7). A logical architecture is an abstraction of a physical architecture and is used to symbolically execute the system functions without implementation constraints [27]. Constraints can be derived by executing the system functionality and are stored in the Logical Architecture sub-model, and act as component requirements when defining the physical architecture. Logical components can have various states. States differ to modes in that they are not functional; they explicitly define the condition of a logical component. The logical component ‘Reflector’ (Figure 8), for example, has associated states, seen in Figure 9.

External Entities

The system itself is defined as the space segment, which is represented in the SEAM by a single spacecraft. An external entity is any entity that is external to, but interacts with, the single-spacecraft space segment. They are defined under the External Entities sub-model, presented in Figure 10. The External Entities sub-model comprises the Ground Segment, the Targets and the Environment. The Ground Segment consists of Ground Stations, capable of receiving downlinked science and/or housekeeping data, and Control Centres, which are also capable of uplinking commands to the spacecraft. The Targets sub-model contains information on the physical science targets throughout the mission. For Earth-observation missions, for example, the longitude and latitude of the targets would be provided. The Environment sub-model allows any environmental factors, such as incident light and thermal energy during eclipse, to be modelled. All external entities are characterised by state machines, in which their effects on the spacecraft can be defined.

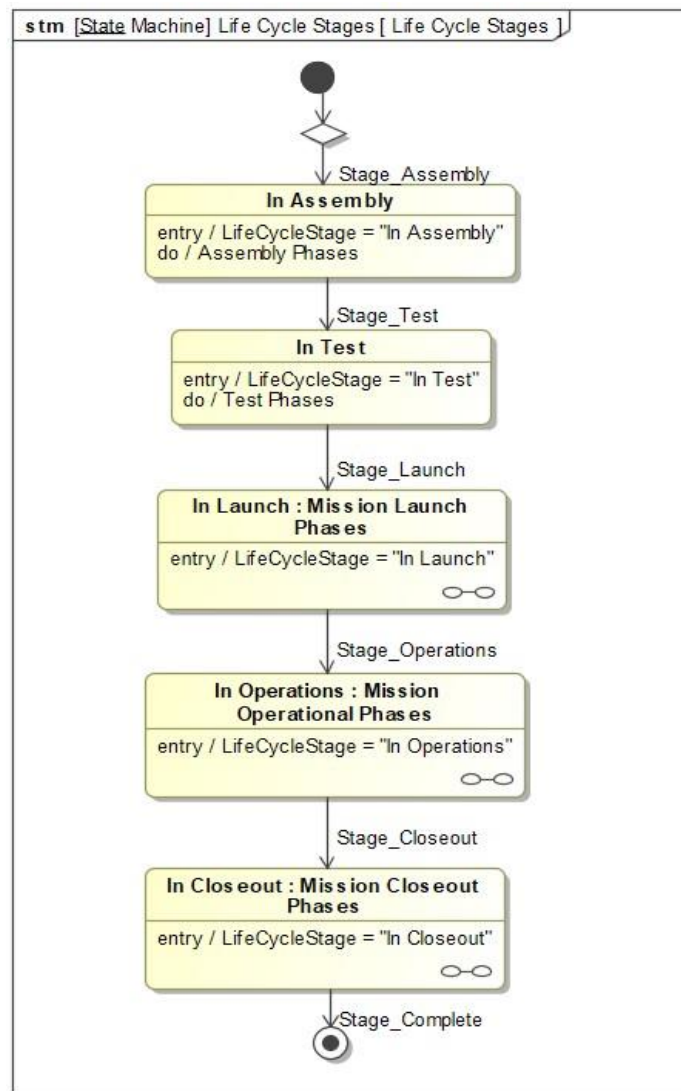


Figure 4: Spacecraft Life Cycle Stages

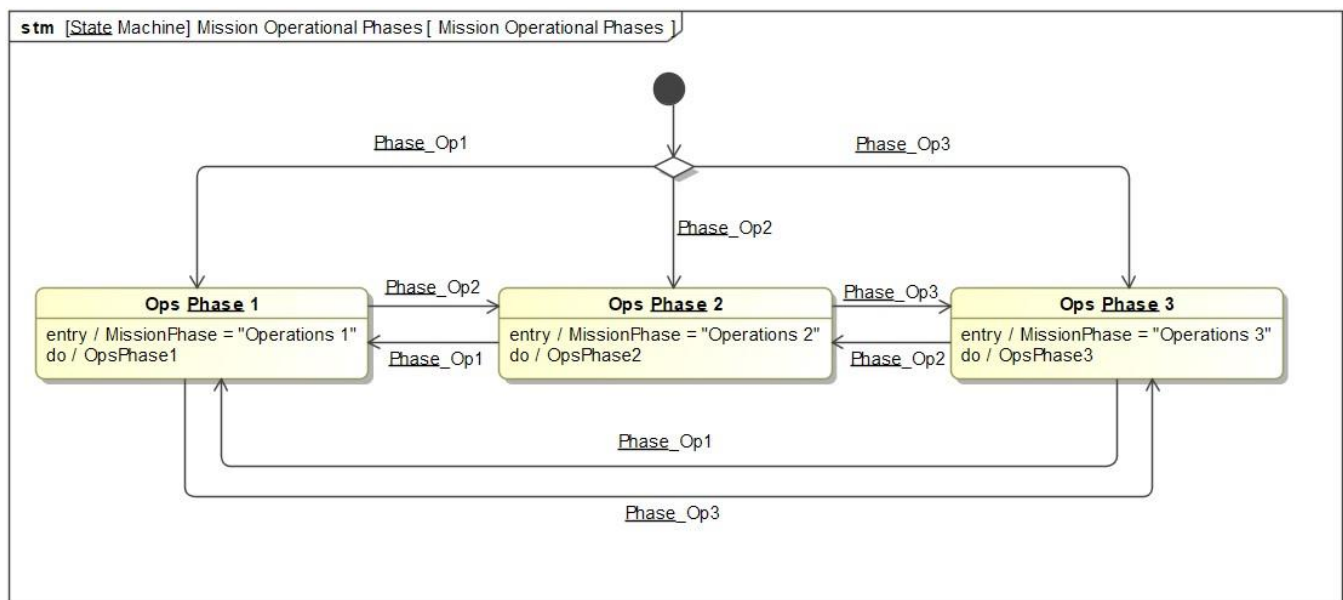


Figure 5: Example of Mission Phases: Mission Operational Phases

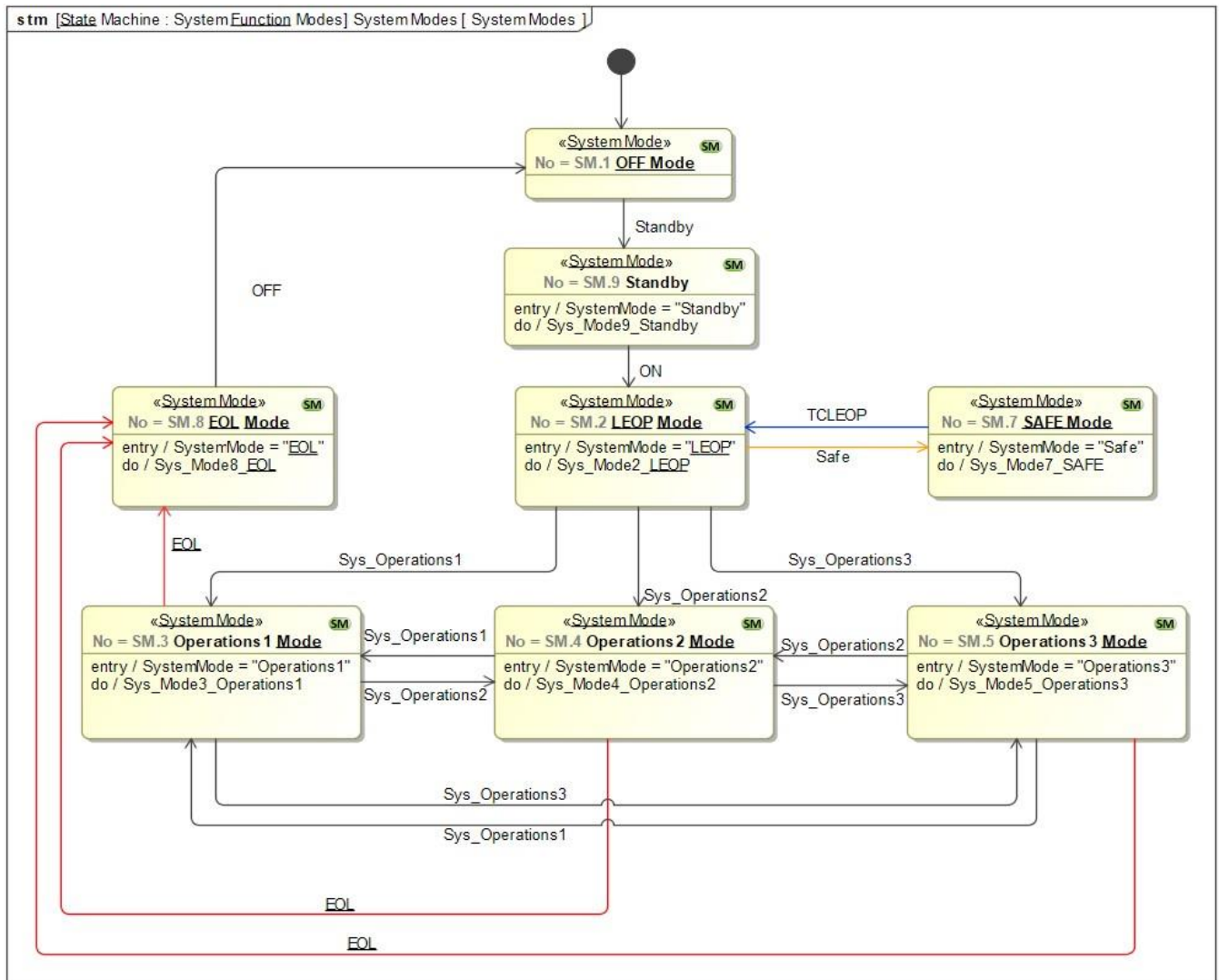


Figure 6: System Mode Diagram

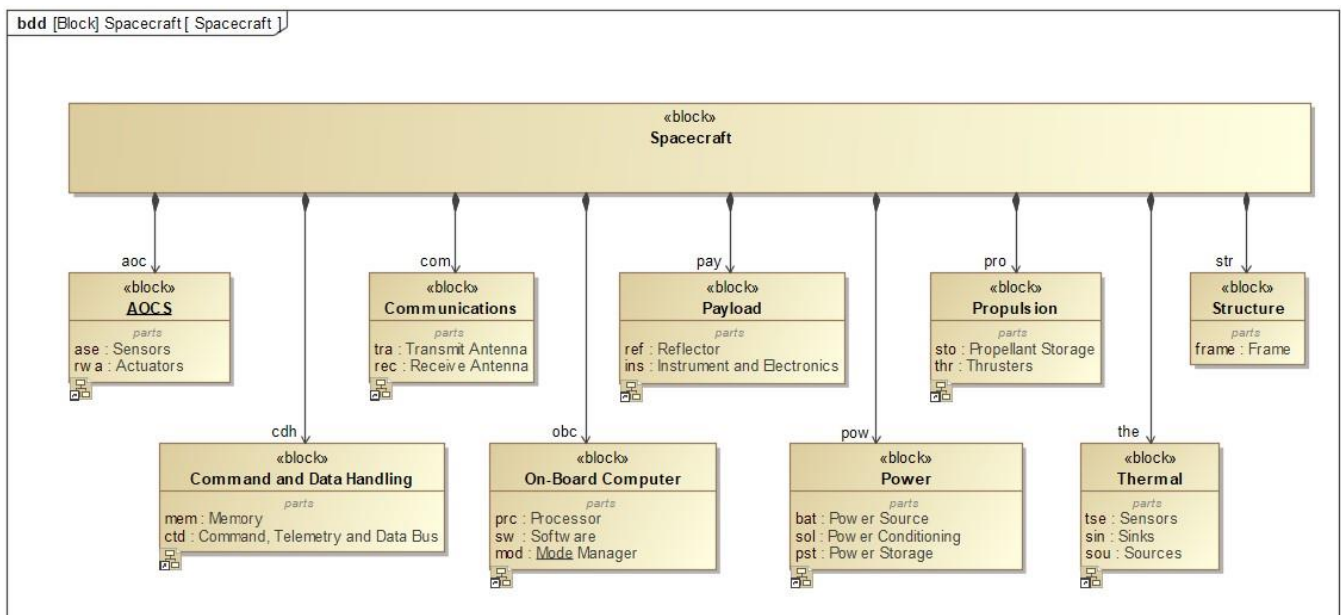


Figure 7: Preliminary Spacecraft Logical Architecture

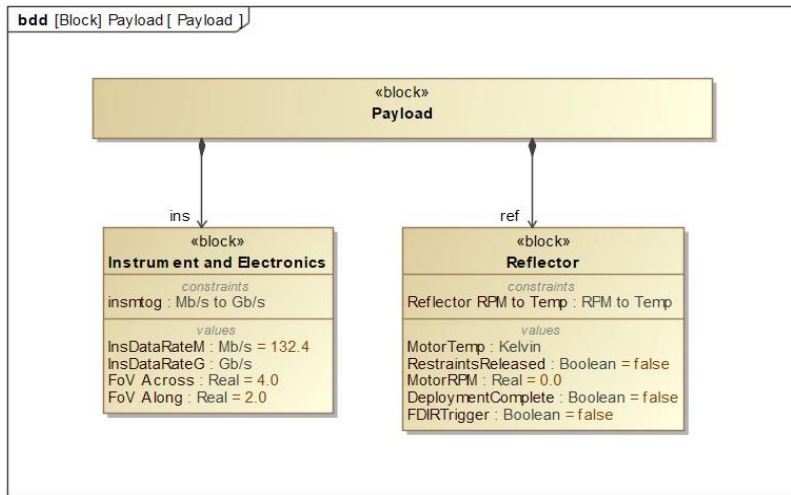


Figure 8: Payload Logical Architecture

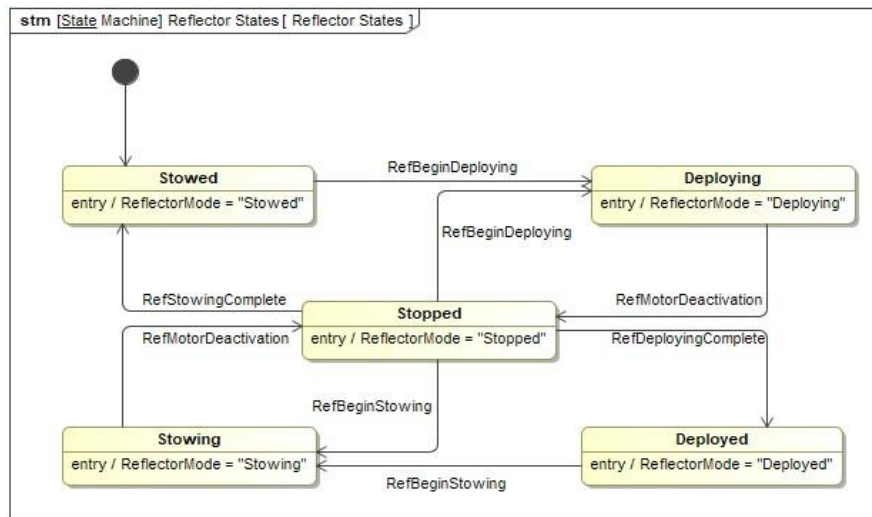


Figure 9: Reflector Logical Component States

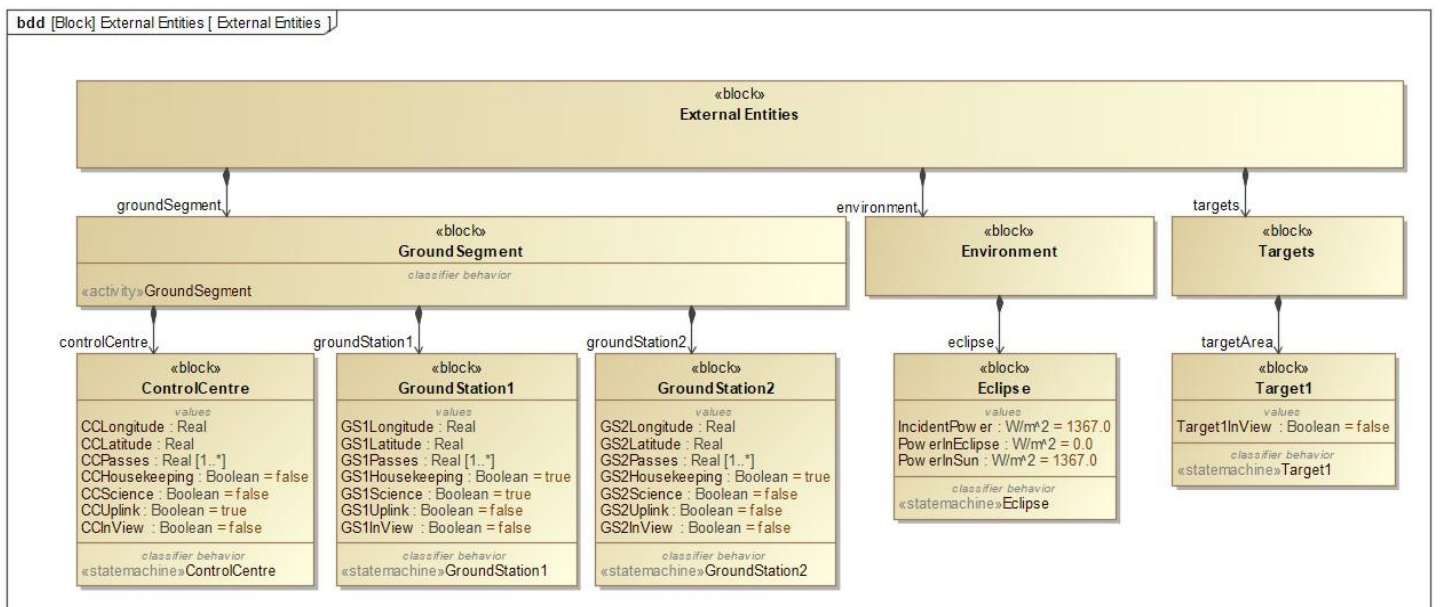


Figure 10: External Entities Sub-Model Structure

STK is specialised mission analysis software dedicated to the modelling of air, space, land and sea operations in simulated or real time [37]. In the SEAM, STK is used to model the orbits of the spacecraft based on the information contained within the Mission Profile and External Entities sub-models. STK can analyse the orbit and accurately determine start and stop times for ground segment passes, target passes and eclipses, thus building up a mission profile that the system functionality can be assessed against.

MATLAB

The SEAM has two uses for MATLAB. First as a bridge between Cameo Systems Modeler and other tools instead of ModelCenter. The SEAM contains a MATLAB script that is capable of reading input information from the Mission Profile sub-model, launching STK, running an STK-based mission analysis, and feeding a concise matrix of the relevant results back into the Mission Profile sub-model.

The SEAM's second use of MATLAB is its capability to analyse mathematical equations and perform simple logic-based operations – even simple mathematics and logic becomes unwieldy very quickly when using SysML activity and parametric diagrams. MATLAB allows for a convenient way of 'outsourcing' all but the simplest mathematical operations.

Microsoft Excel

Microsoft Excel has two uses in the SEAM. On completion of any mission analysis performed in STK, the results can be stored in Excel rather than (or as well as) being transmitted directly into the Mission Profile sub-model. This means that for any subsequent analyses, the initial mission analysis results can be retrieved without needing to rerun the full simulation in STK.

Excel can also be used to store the results of the full SEAM analysis – indeed in the Biomass mass memory use case [10] the results of the simulation, including whether the requirements had been satisfied or not, were stored in Excel.

Simulation Sequence

With the key aspects of the SEAM defined, the simulation sequence can be described with reference to these aspects and the flow of information represented by the arrows in Figure 3. This process assumes that both the mission (stages and phases) and system (functional elements and, where necessary, logical elements) have been defined to an appropriate level of detail and in accordance with the required structure.

1. The core SysML-based model is executed, initialising all behavioural diagrams.

2. The Mission Profile sub-model can either read previously-saved STK results from Excel or use the orbit, ground segment and target definitions defined within it and the External Entities sub-model to launch STK and perform its own analysis via MATLAB. The results are relayed back to the Mission Profile sub-model.

3. With the Mission Profile sub-model now containing information regarding the start and stop times of all ground segment passes, target passes, eclipses, etc., the Mission Profile sub-model can send signals to the Environment and Life Cycle Stages and Mission Phases sub-models. This will update the life cycle stage (Figure 4), mission phase (e.g. Figure 5), and/or the state of the external entities (seen in Figure 10) and specify the duration until the next change.

4. Updating the life cycle stage, mission phase or external entity status will trigger a response from the system – usually a mode transition (Figure 6). For example, if the mission phase transitions to 'Ops Phase 1', the system should respond by transitioning to 'Operations 1 Mode'. If a particular target comes into view in the Target sub-model, this could trigger a transition to 'Operations 3 Mode', for example.

5. Transitioning into a new system mode will trigger the execution of a series of system functions – a functional chain. The functional chain for 'Operations 1 Mode' is presented in Figure 11 – note that the functional chains for all modes

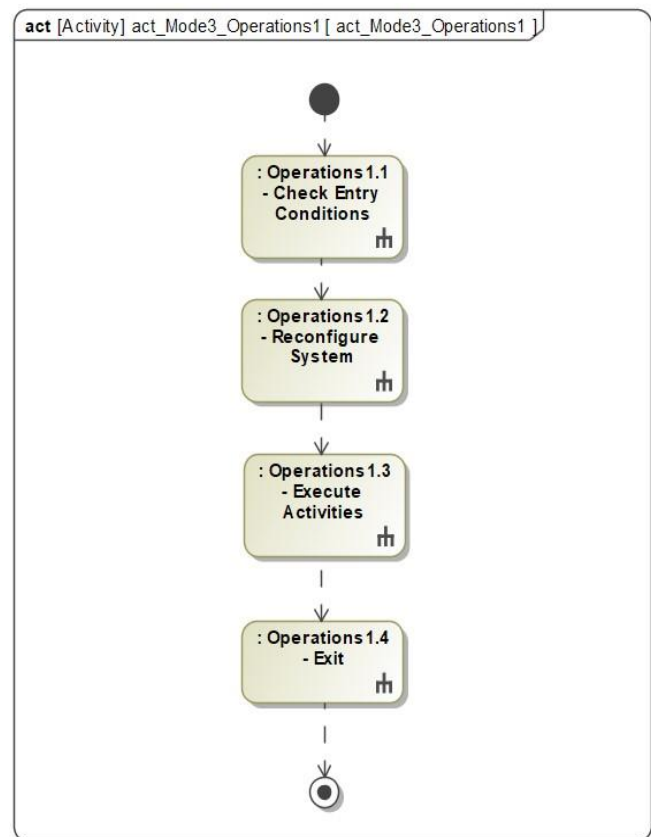


Figure 11: 'Operations 1 Mode' Functional Chain

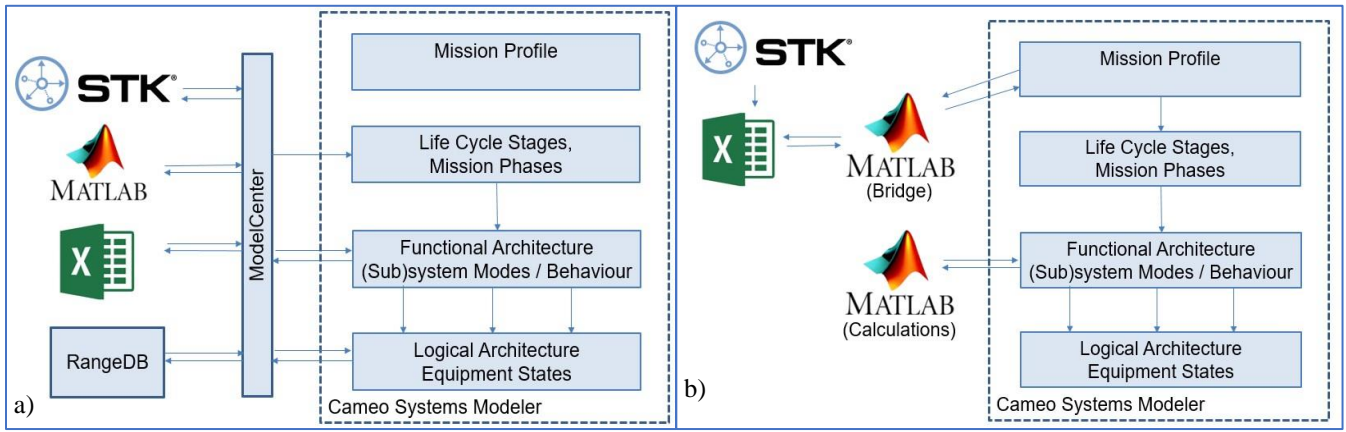


Figure 12a: Structure of the Federated and Executable Models template (used on e.Deorbit).

Figure 12b: Previous Structure of the Spacecraft Early Analysis Model (used on the first Biomass Use Case)

follow this pattern. Each mode is characterised by a series of functions that the system must perform. Each function can consist of multiple levels of subfunctions.

6. Lower level functions may include calculations, which can be performed by MATLAB. They could also include changing the state of the logical components – changing the logical component Reflector from ‘stowed’ to ‘deployed’ (Figure 9), for example. In this case, communication with the Logical Architecture sub-model is required. Furthermore, if a logical component is in an incorrect state, a function may not be possible. It would be impossible to complete the function ‘transmit data to Earth’, for example, if the logical component ‘Transmit Antenna’ was in the ‘stowed’ rather than ‘deployed’ state.

7. On completion of the appropriate system functionality and relevant calculations within that mode, the simulation loops back to Step 2, the mission time progresses accordingly, and the next life cycle stage, mission phase or external entity state transition is triggered.

8. The simulation will close on reaching a predefined mission time, at which point the results of the simulation will be saved to Excel. Examples of possible results include: a timeline of the amount of science data stored onboard the spacecraft throughout the mission; warnings of incomplete functions; duration analysis of a functional chain (e.g. concerning deployment); a proposed communication schedule with a predefined ground station; a summary of which requirements are (not) satisfied, etc. An example of a timeline that can be generated by simulation is presented in Figure 13, taken from [10]. This shows the data stored on-board the Biomass spacecraft during the first three days of its mission.

Summary of Key Features

The evolution of the closed STK loop can be seen by observing two previous versions of the template structure, shown in Figure 12. Figure 12a. displays the structure of the model template used with the Federated and Executable Models approach, and Figure 12b. displays the SEAM structure following the first Biomass use case. Following the first Biomass use case, the need for ModelCenter had been

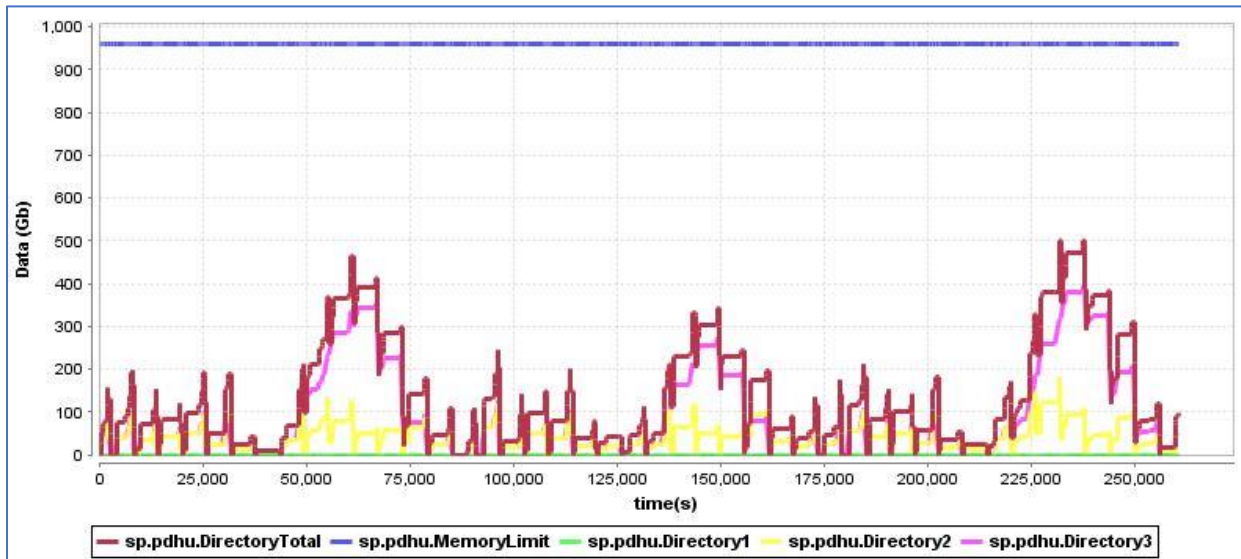


Figure 13: Evolution of Data Stored in the Biomass Spacecraft's Payload Data Handling Unit (PDHU) [10]

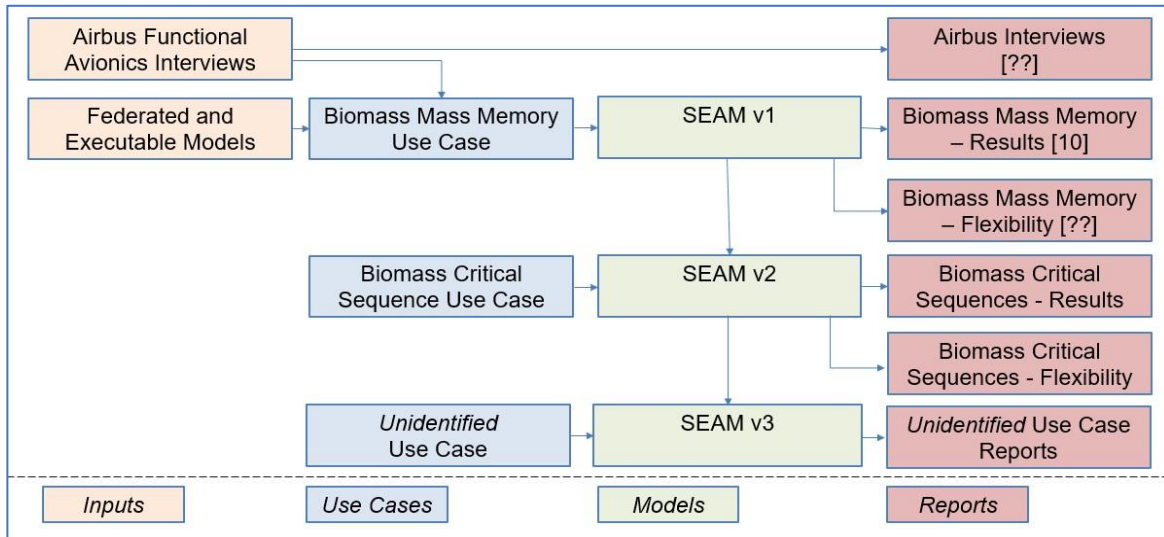


Figure 14: Development Process of the Spacecraft Early Analysis Model

removed by implementing a MATLAB bridge instead, but the STK-based mission analysis still had to be executed independently, with the Excel-based results then being read into the Mission Profile during the system analysis at a later date. Closing the loop, as seen in Figure 3, allows the full analysis, including the STK-based mission analysis, to be defined and executed from the core SysML-based model. Assuming MATLAB was already implemented in the system model, which is the case for the SEAM, this means one fewer tool is required, resulting in cost and compatibility benefits.

Note that the RangeDB tool has not been included in the latest version of the SEAM. The RangeDB tool is a database tool, developed internally by Airbus, to store model parameter values that are used to describe a physical system [19]. For applications of the SEAM, which are in the early stage of design, detailed physical design information is not yet available and so RangeDB is not required – any specific parameter values associated with the model can be stored directly in the Logical Architecture sub-model.

Throughout the development of the SEAM, it has been imperative to maintain the primary purpose of this design information – to provide a clear description of the system under design. This purpose is as it would be using traditional document-based approaches. While the SEAM structures this in a model-based environment to enable its simulation, the design information remains clear and communicable. Each diagram can still be viewed as a definition of some aspect of the mission (e.g. life cycle stages, Figure 4) or system (e.g. system mode diagram, Figure 6) independently.

The SEAM is built on repeatable patterns to improve the ease with which additional features can be added to the template. The functional chain presented in Figure 11 can be used to introduce additional modes. This exercise has been performed and its efficiency reviewed in [35]. There is also a standard pattern by which additional MATLAB scripts can be called from the core model, thus providing an easy way to

maintain, update, add and remove mathematical analyses of the system. Another example is the pattern by which further external entities can be added and their influence on the system accounted for.

5. DISCUSSION

In this section the benefits and limitations of the current SEAM structure as presented in this paper are discussed in more detail, and potential areas for improvement are identified.

The SEAM has undergone ‘development by use case’, in which its simulation capabilities and structure have been developed to accommodate the needs of a use case and subsequently generalised with the objective of producing a template that is general enough to applied to a variety of missions. This methodology can be contrasted with the development of the CRM, which has undergone a more ‘top-down’ development approach [24]. Other differences between the development approaches adopted for the SEAM and the CRM are that the SEAM is heavily based in a single industrial organisation, Airbus, and that the SEAM has been developed to accommodate a general ‘spacecraft’ as opposed to the more specific CRM focus of CubeSats. Approaching the development of the SEAM in this way has introduced some issues that must be addressed going forward.

Development of the SEAM via repeated use case application and generalisation will only produce a versatile template if a variety of use cases are used. Both use cases used in its development so far are products of the Biomass mission; a low-Earth orbit Earth-observation mission. While efforts have been made to keep the SEAM as general as possible, its versatility cannot be developed and demonstrated until it is applied to other missions. Possible considerations include interplanetary missions, crewed missions, probes and rovers. Even if the scope of the SEAM is limited to Earth-observation spacecraft, use cases from Earth-observation spacecraft other than Biomass must be modelled.

With the development of the SEAM has come increased complexity, and this must be managed carefully. As discussed in the previous section, separation between descriptive diagrams and analytical diagrams must be maintained. The SEAM must ensure that system definition diagrams are not contaminated with elements that are not necessary to describe the system, but which are necessary to produce the simulation. This may require the production of behavioural diagram ‘twins’ – two mutually consistent representations of some aspect of the model – one with all analytical model elements hidden (for system definition) and another with these displayed.

Utilising the SEAM has provided useful insights into both of the Biomass use cases. The Biomass Mass Memory use case observed improvements with regards to the communicability, consistency and navigability of the design information, the level of analysis that it makes possible, and its flexibility to typical systems engineering changes (e.g. addition of a requirement, late solution change, etc.) [35]. Quantifying these benefits, however, has proven difficult. The objective of this project is to introduce a model-based template to be used on space-based missions and the benefits. More work needs to be done on how these benefits can be quantified.

6. NEXT STEPS

Figure 14 provides an overview of the development of the SEAM and highlights the future direction of the project. The next steps will be to address the points raised in the previous section.

The SEAM will be applied to a new use case on a different, non-Earth-observation, mission. In this way its flexibility can be assessed, new features can be added as required, and patterns to be followed when applying the SEAM to new missions can be created and refined as necessary.

As the SEAM is applied to new use cases, clear instructions on the production of system ‘definition’ and system ‘analysis’ diagrams, and behavioural diagram ‘twins’, will be produced. This will be done with the aim of managing complexity and ensuring that system definition diagrams are not contaminated with model elements required only for system analysis.

The use of metrics to evaluate the application of the SEAM to a mission over traditional DBSE methods will be investigated. Feedback regarding the SEAM from the same Functional Avionics engineers that were involved in the initial Airbus interviews used to define the project direction may be one method of measuring how well their needs have been addressed.

7. CONCLUSIONS

This paper has outlined the development, current status and future direction of the Spacecraft Early Analysis Model (SEAM). The specific need for the SEAM was identified

through interviews with Airbus engineers working in Functional Avionics, who highlighted in particular the need for a template model framework and the capability to perform early validation of the concept of operations. The Biomass mission, under development by Airbus for the European Space Agency (ESA), has yielded two use cases within the realm of Model-Based Avionics Engineering (MBAE). The current version of the SEAM, as presented in this paper, looks to address the concerns raised by the Airbus engineers and has undergone ‘development by use case’. The SEAM pulls together different definition and analysis tools that would otherwise be operating independently, connecting them to yield an integrated system model capable of simulation and analysis.

Limitations arising from the structure of the SEAM have been identified and will be addressed as the ‘development by use case’ continues. The flexibility and robustness of the SEAM in terms of typical systems engineering project changes has been investigated, and similar investigations into the benefits of its application to other missions will be carried out. Subsequent versions of the SEAM will be developed as the second Biomass use case is completed and a third, currently unspecified use case is undertaken. Alongside this work will be the continuous assessment of the SEAM’s relevance against the needs of Functional Avionics engineers within both Airbus and the wider systems engineering community.

ACKNOWLEDGEMENTS

The views expressed in this paper do not necessarily represent those of Airbus. The work performed here was funded under EPSRC grant number 16000151 supported by Airbus. The authors would like to acknowledge support from Alexandre Cortier, Stephane Estable, Thomas Fenal, Joanna O’Rourke, Antonio Prezzavento, Alain Rossignol. The authors would also like to thank No Magic for their provision of a Cameo Systems Modeler™ (19.0) Evaluation Licence, under which this work was completed.

REFERENCES

- [1] D. R. Wibben and R. Furfaro, “Model-Based Systems Engineering approach for the development of the science processing and operations center of the NASA OSIRIS-REx asteroid sample return mission,” *Acta Astronautica*, vol. 115, pp. 147–159, 2015.
- [2] K. M. Gough and N. Phojanamongkolkij, “Employing Model-Based Systems Engineering (MBSE) on a NASA Aeronautics Research Project: A Case Study,” in *AIAA Aviation Technology, Integration, and Operations Conference*, Atlanta, GA, 2018.
- [3] R. S. Kalawsky *et al.*, “Bridging the gaps in a model-based system engineering workflow by

- encompassing hardware-in-the-loop simulation,” *IEEE Systems Journal*, vol. 7, no. 4, pp. 593–605, 2013.
- [4] B. London and P. Miotto, “Model-Based Requirement Generation,” in *IEEE Aerospace Conference*, Big Sky, MT, 2014.
- [5] M. Bozzano *et al.*, “Spacecraft early design validation using formal methods,” *Reliability Engineering & System Safety*, vol. 132, pp. 20–35, 2014.
- [6] L. Anderson *et al.*, “Enterprise Modeling For CubeSats,” in *IEEE Aerospace Conference*, Big Sky, MT, 2014.
- [7] S. Chhaniyara, C. Saaj, M. Althoff-Kptzias, I. Ahrens, and B. Maediger, “Model Based System Engineering for Space Robotics Systems,” in *ESA Symposium on Advanced Space Technologies in Robotics and Automation*, Noordwijk, Netherlands, 2011.
- [8] Y. Jarraya, A. Soeanu, M. Debbabi, and F. Hassaïne, “Automatic verification and performance analysis of time-constrained SysML activity diagrams,” in *IEEE International Symposium and Workshop on Engineering of Computer Based Systems*, Tucson, AZ, 2007.
- [9] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, “Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission,” in *IEEE Aerospace Conference*, Big Sky, MT, 2014.
- [10] J. Gregory, L. Berthoud, T. Tryfonas, and A. Prezzavento, “Early Validation of the Data Handling Unit of a Spacecraft Using MBSE,” in *IEEE Aerospace Conference*, Big Sky, MT, 2019.
- [11] S. J. Kapurch, “NASA Systems Engineering Handbook,” NASA Headquarters, Washington, D.C. NASA/SP-2007-6105, 2007.
- [12] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. London: Addison-Wesley, 2014.
- [13] INCOSE, “What is Model Based Systems Engineering?,” *INCOSE UK zGuide*, 2015. [Online]. Available: https://incoseonline.org.uk/Documents/zGuides/Z9_model_based_WEB.pdf. [Accessed: 12-Sep-2019].
- [14] A. L. Ramos, J. V. Ferreira, and J. Barceló, “Model-based systems engineering: An emerging approach for modern systems,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 1, pp. 101–111, 2012.
- [15] Object Management Group, “An OMG Systems Modeling Language TM Publication OMG Systems Modeling Language v1.5,” Needham, MA, 2017-05-01, 2017.
- [16] M. Brumbulli, E. Gaudin, A. Cortier, and A. Rossignol, “System Optimization : A Use Case in the Space Domain,” in *9th European Congress on Embedded Real Time Software and Systems and Systems*, Toulouse, France, 2018, p. 9.
- [17] S. Estable *et al.*, “Systems Modelling and Simulation of the ESA e.Deorbit Space Debris Removal Mission,” in *NAFEMs*, Stockholm, Sweden, 2017.
- [18] R. Biesbroek, “ESA’s e.Deorbit Mission and its Roadmap to Active Debris Removal,” in *5th Challenges in European Aerospace (CEAS) Conference*, Delft, Netherlands, 2015, no. 153, pp. 1–14.
- [19] S. Estable, “Application of the ‘Federated and Executable Models’ MBSE Process to Airbus Orbital Servicing Missions,” in *Phoenix Integration International Users’ Conference*, Annapolis, MD, 2018.
- [20] RHEA GROUP, “Concurrent Design Platform (CDP) 4 | www.rheagroup.com.” [Online]. Available: <https://www.rheagroup.com/cdp>. [Accessed: 08-Oct-2019].
- [21] MathWorks, “MATLAB - MathWorks - MATLAB & Simulink.” [Online]. Available: <https://uk.mathworks.com/products/matlab.html>. [Accessed: 08-Oct-2019].
- [22] Phoenix Integration, “ModelCenter MBSE | Model Based Systems Engineering | Phoenix Integration.” [Online]. Available: <https://www.phoenix-int.com/product/mbse/>. [Accessed: 04-Oct-2019].
- [23] A. Rossignol, S. Estable, A. Cortier, and D. Thomas, “Model-based Avionics Roadmap and Case Studies,” in *ESA Workshop on Avionics, Data, Control and Software Systems*, Noordwijk, Netherlands, 2016.
- [24] D. Kaslow, “Developing a CubeSat MBSE Reference Model - Interim Status #3,” in *IEEE Aerospace Conference*, Big Sky, MT, 2017.
- [25] D. Kaslow, B. Ayres, I. Curtis, and B. Shiotani, “CubeSat MBSE Reference Model - Application in the Concept Lifecycle Phase.pdf,” in *AIAA SPACE 2015 Conference and Exposition*, Pasadena, 2015.
- [26] S. C. Spangelo *et al.*, “Model based systems engineering (MBSE) applied to Radio Aurora Explorer (RAX) CubeSat mission operational scenarios,” in *IEEE Aerospace Conference*, Big Sky, MT, 2013.
- [27] D. Kaslow and A. Madni, “Validation and Verification of MBSE-compliant CubeSat Reference Model,” in *Annual Conference on Systems Engineering Research*, Springer, Cham, Redondo Beach, CA, 2017, pp. 1–10.

- [28] L. Lindblad, "Data-Driven Systems Engineering. Turning MBSE into Industrial Reality," in *ESA International Systems & Concurrent Engineering for Space Applications Workshop*, Glasgow, UK, 2018.
- [29] S. Jenkins, "Is Systems Engineering Really Engineering," in *ESA International Systems & Concurrent Engineering for Space Applications Workshop*, Glasgow, UK, 2018.
- [30] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," *Systems Engineering*, vol. 21, pp. 172–190, 2018.
- [31] S. Feo-Arenis, "SAVOIR-TN-003: Model-Based Avionics Roadmap," Iss 1, Rev 0. ESTEC, ESA. Noordwijk, The Netherlands, 2018.
- [32] J. Gregory, L. Berthoud, T. Tryfonas, A. Rossignol, and L. Faure, "The long and winding road: MBSE adoption for functional avionics of spacecraft," *Journal of Systems and Software*, vol. 160, p. 110453, 2019.
- [33] M. Arcioni *et al.*, "The Biomass Mission, Status of the Satellite System," in *IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC, Canada, 2014.
- [34] European Space Agency, "Report for Mission Selection, ESA SP-1324(1)," Mission Science Division, ESA-ESTEC, Noordwijk, The Netherlands, 2012.
- [35] J. Gregory, L. Berthoud, T. Tryfonas, A. Prezzavento, and L. Faure, "Investigating the Flexibility of the MBSE Approach to the Biomass Mission," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. (In press), 2019.
- [36] No Magic, "Cameo Systems Modeler." [Online]. Available: <https://www.nomagic.com/products/cameo-systems-modeler>. [Accessed: 07-Oct-2019].
- [37] AGI, "STK. Engineering Tools | AGI." [Online]. Available: <https://www.agi.com/products/engineering-tools>. [Accessed: 04-Feb-2019].
- [38] European Space Agency, "Key components of tree-counting Biomass radar cleared for space / Space Engineering & Technology / Our Activities / ESA." [Online]. Available: http://www.esa.int/Our_Activities/Space_Engineering_Technology/Key_components_of_tree-counting_Biomass_radar_cleared_for_space. [Accessed: 08-Oct-2019].

BIOGRAPHY



Joe Gregory is a PhD student at the University of Bristol where he is investigating the application of Model-Based Systems Engineering (MBSE) to spacecraft functional avionics in collaboration with Airbus Defence and Space. He received his MEng degree in Aeronautical Engineering from the University of Bristol in 2014. The focus of his current work is early functional validation in the context of Airbus Defence and Space functional avionics.



Professor Lucy Berthoud has a Master's in Mechanical Engineering from the University of Bristol and a PhD in Space Physics from Sup'Aero in France. She did Post-doctoral fellowships at the European Space Agency and NASA Johnson Space Centre, before going to work for British Aerospace Space Systems. In 2009 she started working for the University of Bristol where she specialises in systems engineering for spacecraft.



Dr Theo Tryfonas MBCE CITP, FRSA is a Reader in Smart Cities with the Department of Civil Engineering of the University of Bristol, with a background in systems engineering, software development and cyber security. His research expertise includes secure and resilient operation of Internet of Things (IoT) applications, privacy in mobile computing, energy efficient deployments of wireless sensor networks and open data architectures for smart buildings/infrastructure.



Ludovic Faure is a space professional who has worked for CNES, ESA and now Airbus Defence and Space. Starting as a software engineer, he has specialised in Model Driven Architecture (MDA) using Eclipse EMF, and collaborated with the project Topcased (UML/SysML editor) for the CNES. Passionate about the Space domain, he became Simulation Officer for the Galileo Project, and was in charge of training the ESOC Mission Control Team for 7 Galileo launches. Ludovic joined Airbus in 2017 and is now working as Operations/FDIR Architect for the mission Exomars. He also acts as the coordinator for R&D Activities for the Functional Avionics domain in Stevenage, UK.